

Monte Carlo Production Management at CMS

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2015 J. Phys.: Conf. Ser. 664 072018

(<http://iopscience.iop.org/1742-6596/664/7/072018>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 131.215.70.231

This content was downloaded on 22/04/2016 at 22:26

Please note that [terms and conditions apply](#).

Monte Carlo Production Management at CMS

**G Boudoul¹, G Franzoni², A Norkus^{2,3}, A Pol², P Srimanobhas⁴
and J-R Vlimant⁵ - for the Compact Muon Solenoid collaboration**

¹ U. C. Bernard-Lyon I, 43 boulevard du 11 Novembre 1918 69622 Villeurbanne, France

² CERN, PH CMG-CO; CH-1211 Geneva 23 Switzerland

³ University of Vilnius, Universiteto g. 3, Vilnius 01513, Lithuania

⁴ Chulalongkorn University, 254 Phayathai Road, Pathumwan, Bangkok 10330, Thailand

⁵ California Institute of Technology, 1200 East California Boulevard, Pasadena, CA 91125, United States

E-mail: giovanni.franzoni@cern.ch

Abstract. The analysis of the LHC data at the Compact Muon Solenoid (CMS) experiment requires the production of a large number of simulated events. During the RunI of LHC (2010-2012), CMS has produced over 12 Billion simulated events, organized in approximately sixty different campaigns each emulating specific detector conditions and LHC running conditions (pile up). In order to aggregate the information needed for the configuration and prioritization of the events production, assure the book-keeping of all the processing requests placed by the physics analysis groups, and to interface with the CMS production infrastructure, the web-based service Monte Carlo Management (McM) has been developed and put in production in 2013. McM is based on recent server infrastructure technology (CherryPy + AngularJS) and relies on a CouchDB database back-end. This contribution covers the one and half year of operational experience managing samples of simulated events for CMS, the evolution of its functionalities and the extension of its capability to monitor the status and advancement of the events production.

1. Introduction

The Compact Muon Solenoid (CMS) experiment [1] is an omni-purpose detector operating at the Large Hadron Collider [2] at CERN. The central feature of the CMS apparatus is a superconducting solenoid of 6 m internal diameter, providing a magnetic field of 3.8 T. Within the superconducting solenoid volume are a silicon pixel and strip tracker, a lead tungstate crystal electromagnetic calorimeter (ECAL), and a brass/scintillator hadron calorimeter. Muons are measured in gas-ionization detectors embedded in the steel flux return yoke outside the solenoid. In addition, the CMS detector has extensive forward calorimetry. The first level of the CMS trigger system, composed of custom hardware processors, uses information from the calorimeters and muon detectors to select the most interesting events in a fixed time interval of less than 4 μ s. The High Level Trigger (HLT) computer farm further decreases the event rate from around 100 kHz to around 0.5 kHz, before data storage.

The analysis of the LHC data at CMS experiment requires the production of a large number of simulated events. During the RunI of LHC (2010-2012), CMS has produced over 12 Billion simulated events, organized in approximately sixty different campaigns each emulating specific



detector and LHC running conditions, such as multiple interactions per bunch crossing (pileup), beam-spot shape and position, etc [4]. Such vast scale production of simulated events has continued during the first LHC long shutdown (LS1): at first in preparation of the updated offline data simulation and reconstruction software, and later, from the autumn of 2014, to produce simulated the events which will be used in the analysis of the LHC RunII data.

This paper describes the procedures and infrastructure used by CMS to produce simulated events, and is structured as follows: the roles involved in the procedures are outlined first, followed by the description of the web-based platform (McM) used to aggregate and submit the processing requests to the CMS computing infrastructure. The second part of the paper is dedicated to a novel monitoring infrastructure (pMp) and the outlook.

2. Monte Carlo Management: Roles and McM Platform

The roots of the CMS collaboration rest on working teams, the so called level two groups, each charged with a set of goals and deliverables. Detector (e.g. the electromagnetic calorimeter ECAL detector performance group), physics object (muon physics object group) or analyses (exotica analysis group) teams are three examples of level two groups we will deal with in this paper. Any such group needs simulated datasets, either to qualify the performance they are responsible for or to produce published results.

Several universities and research laboratories participating to CMS share part of their computing facilities with the collaboration, which uses them either for official data processing or to serve user jobs for analysis [3]. The production of simulated events employed by CMS for public results is carried out using the shared storage and processing resources available at the regional Tier-1 and Tier-2 centres around the world. The organisation of such production is the responsibility of the Monte Carlo Coordination Meeting (MCCM) team, part of the Physics Performance and Datasets coordination area. The MCCM team collects inputs from the Physics Coordination leaders and from all the level two groups, draws a plan to deliver the agreed datasets exploiting the computing and time resources available at best, and submits detailed production request to the operation team of the Computing coordination area. Three key roles take part in this process:

- generator contact: collects the needs for simulated datasets from within the detector or physics group she/he represents, proposes them to the MCCM for production and composes and tests the configuration of the physics event generator to provide the desired collision type and final state;
- generator convener: scrutinises and approves the event generator configurations proposed by the generator contact;
- request manager: defines and configures production campaigns and submits requests to the production infrastructure.

Since mid 2014, the Monte Carlo Management (McM) web-based platform is the service used by the generator contacts, generator conveners and request managers to exchange information and submit the production requests. McM aggregates the needs for generation and processing of simulated events, holds the configuration of the processing jobs, provides their bookkeeping and prioritisation and submits the requests of events production to the computing resources. The McM front end web interface is based on recent server infrastructure technology (CherryPy + AngularJS), while the back-end database uses CouchDB. As represented in figure 1, these are the services McM interacts with:

- wmAgent [5]: McM submits requests for event production to the request manager infrastructure of the CMS computing project, by providing the CMSSW (the offline software framework used by CMS) job configuration and other key parameters, for instance the

number of event to be processed, the alignment and calibration scenario, the name of the dataset to be produced. The wmAgent framework acts on the processing requests delivered to the request manager and assigns them to a pool of CMS computing resources (at Tier-1 and Tier-2 centres), determines the splitting of events across the jobs, handles recovery attempts of processing failures and publishes completed datasets to dbs [6];

- statsDb: a CouchDB-based database which, for any submitted processing request (MC or data alike), dynamically aggregates the status and growth of datasets, during and after their production stage;
- production Monitoring platform (pMp): pMp dynamically aggregates status of processing requests and size of the output dataset by importing information from McM and dbs/statsDb, and provides monitoring plots; its statistics which can be browsed, either for a single request or for a set of requests form the same chained campaign or the same campaign. A more detailed description of pMp is available in section 4.

3. Requests Definition and Handling

Processing requests for events simulation are split across campaigns and classified according to the simulation step they execute. Depending on the physics goal a given simulated dataset is intended to achieve, the production of events ready for analysis can comprise up to six different steps, see figure 2:

- wmLHE: simulation of the hard event by specialised event generator programs, resulting in events written in LHE format [7];
- generation: hadronization of the hard event, resulting in the kinematic description of the final state particles which propagate from the collision point towards the CMS detector;
- simulation: interaction of the final state particles with the sensitive volumes and material budget of the CMS apparatus, resulting in a collection of time-stamped energy deposits for every sensor;
- digitisation: emulation of the CMS front-end electronics, resulting in digital information formatted as if it was raw data acquired by the real experiment;
- reconstruction: unpacking of the raw data and construction of physically interpretable objects such as tracks, clustered calorimetric deposits and particle flow candidates [8]; the sequence of algorithms of this step are the same for simulated and for real events;

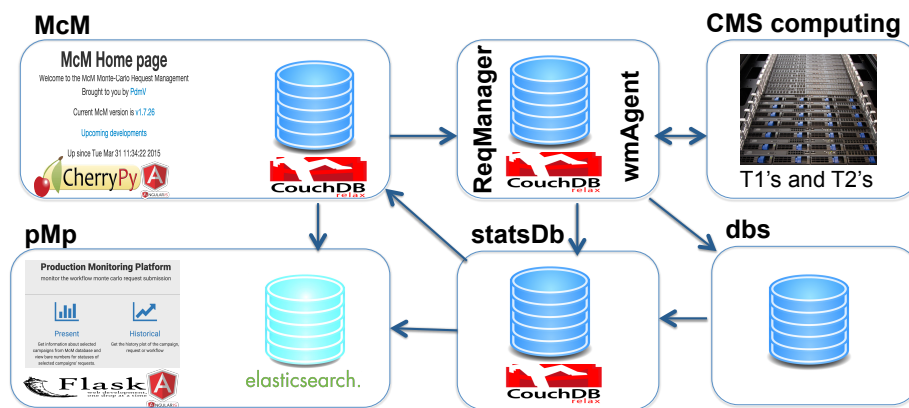


Figure 1. The Monte Carlo Management (McM) platform and the services it interacts with; see text for a detailed description.

- miniAOD: reduction of the reconstruction output to a minimal set of variables, sufficient to carry out the majority of the physics analysis;

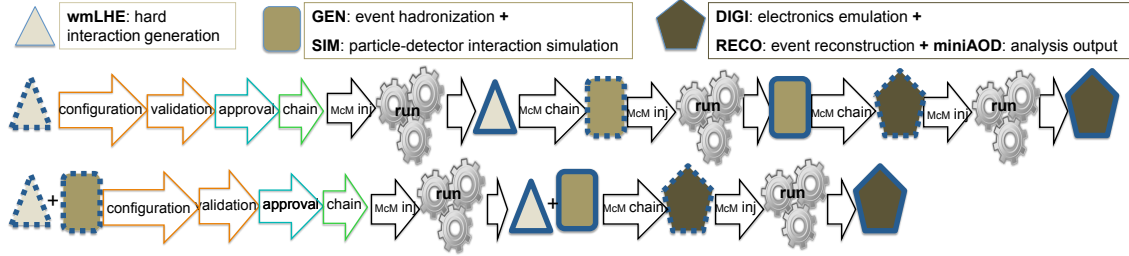


Figure 2. Types of event production requests at CMS, and sequence of steps to obtain simulated events ready for analysis: an example of normal sequence (top) and a task-chained sequence (bottom) are reported. See the text for more details.

The capability of factoring the production steps offered by the CMSSW framework brings several key advantages, but comes with a cost of complexity. Among the advantages the fact that the earlier steps of the chain (typically up to the simulation) can be executed ahead of time, while the subsequent steps (notably the reconstruction) are still being developed and validated; in addition, different decay channels (in the generation step) or scenarios of alignment and calibration (in the digitisation) can be simulated branching from the same events different dataset versions, with no need of repeating the previous common steps. The complexity arises from the need of defining and submitting to production multiple processing requests for any given dataset needed for analysis: as shown in figure 2 (top), when all the six production steps are necessary, grouping is possible such that only three processing requests need to be placed.

The configuration of event processing requests in McM is structured around four main entities, which are at the basis of the architecture of the service and are intended to facilitate and logically organise large sets of interdependent requests:

- Campaign: it is a set of requests for the same physics analysis goal, sharing software release, beam energy and event processing configuration. Currently there are 83 campaigns defined in McM.
- Flow: it connects a dataset produced in a campaign to the subsequent request which is part of the next campaign (e.g. the output of the simulation step is flown as input to the campaign executing digitisation and reconstruction); flows implement the modifications to the baseline processing configuration of a given the campaign to implement a specific processing scenario, like for instance the pile up or the event content. Currently there are 178 flows defined in McM.
- Chained campaign: a sequence of campaigns connected by flows determining the succession of processing steps and campaigns which are needed to deliver datasets for analysis. Currently there are 317 chained campaign defined in McM.
- Chained request: a concrete set of processing requests (currently there are about 18000 chained requests in McM), starting from a root request (e.g. a given generated event topology) and going through the steps of a chained campaign. Currently there are about 10000 chained requests defined in McM.

The request managers are in charge of constructing campaigns, flows and chained campaigns: these three constitute the back-bone of the McM design. Once they are in place, McM takes

care automatically of a large set of mechanical steps needed for the submission of processing request, among which: the validation of the generator configuration (by triggering the execution of a test job on a limited number of events), the measurement of the generator efficiencies and CPU time per event and the actual submission to the request manager - see figure 2 (top) for a detailed break-down. The actual submission of processing request to the computing operations group consists of one chained request per desired analysis dataset.

In 2014 a novel kind of processing request has been made available to McM allowing the submission to production of multiple requests in the same workflow, saving labor and shortening the time needed to deliver datasets. As shown in figure 2 (bottom), the current application of such novel approach has allowed the wmLHE, generation and simulation steps to be executed in a 'task-chained' request at the same processing site.

With the pile up scenarios simulated for RunII, the digitisation step can be reliably executed only at the at Tier-1 centres, which limits the scope of task-chaining to the set of simulation steps which can be executed at the same site, namely: from wmLHE to simulation (which Tier-2's can handle) and from digitisation to miniAOD (which can be processed only by Tier-1's). Thanks to the ongoing development of a new pileup simulation technique (premixing, see [9]) which lowers the i/o requirement of the digitisation step by more than one order of magnitude, also the major Tier-2 centres, besides the Tier-1's, will be capable of executing such step. Once the premixing technique will be usable in production, task-chaining multiple requests in a single workflow to run at a single Tier-1 or Tier-2 site (in principle from wmLHE all the way to miniAOD) will bear considerable benefits to the efficiency and delivery time of the CMS simulated event production.

4. Requests Monitoring: the pMp platform

McM offered a suit of monitoring plots from the start, which proved very useful and created the use case for their own expansion. The interactive production of such graphs used the same database back-end and the processing resources needed by the McM service itself. To assure that the service availability would not be jeopardised by excessive traffic, the access to the monitoring had to be restricted to the production managers, generator convenors and a few other users.

The production Monitoring platform (pMp) is a new service recently deployed in production at CMS to expand the capabilities of monitoring the advancement of simulated events production and the MCCM operations. pMp has been designed to expand the suit of monitoring plots and allow also the generator contacts and any CMS users to follow the progress of requests/campaigns; the service is deployed on an independent server and uses a back-end database index (elasticsearch) independent from McM which gets updated at regular intervals by retrieving information for monitoring from the McM and statsDb; such structures provides increased search performance and assures the factorisation of monitoring load from McM operation.

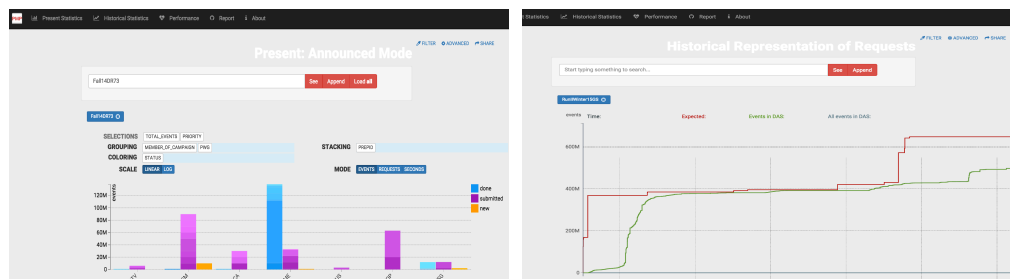


Figure 3. pMp service: present snapshot (left) and historical view (right).

These are the key functionalities offered by pMp

- has access to the requests and campaign information from McM and to the historical advancement of all requests
- allows the monitoring of single requests, chained campaigns or full campaigns
- offers: 1) present snapshot view, where number of events/requests or processing time can be shown in a large set of configurable histograms, and 2) a historical view, where the number of expected and produced events are plotted as a function of time; see figure 3
- the present view can show static values as defined in McM or update them with live information as a request is completed
- the entries displayed in either view can be filtered by variables such as the status of the processing request, the level two group which placed the request, or the priority

5. Summary

In this paper we've described the procedures and tools used by the CMS collaboration to manage the production of simulated events. Three key roles are involved in the process of harvesting requirements from the level two groups of CMS and turning them into an organised set of fully specified processing request to be submitted to the computing infrastructure: the generator contact, generator convenor and request manager. The McM and pMp web-bases services have been described, which take care, respectively, of structuring and bookkeeping large set of processing requests and of monitoring the Monte Carlo production process.

References

- [1] CMS Collaboration, *The CMS experiment at the CERN LHC*, JINST 3 (2008) S08004, doi:10.1088/1748-0221/3/08/S08004.
- [2] Lyndon Evans and Philip Bryant, *LHC Machine*, 2008 JINST 3 S08001, doi:10.1088/1748-0221/3/08/S08001.
- [3] CMS Collaboration, *CMS computing : Technical Design Report*, 2005, CERN-LHCC-2005-023.
- [4] G Franzoni *Data preparation for the Compact Muon Solenoid experiment* Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2013 IEEE, doi: 10.1109/NSSMIC.2013.6829574
- [5] E Fajardo et al for the CMS collaboration *A new era for central processing and production in CMS* 2012 J. Phys.: Conf. Ser. 396 042018 doi:10.1088/1742-6596/396/4/042018
- [6] Valentin Kuznetsov et al for the CMS collaboration *The CMS DBS query language* 2010 J. Phys.: Conf. Ser. 219 042043 doi:10.1088/1742-6596/219/4/042043
- [7] J Alwall et al *A standard format for Les Houches Event Files* 2010 Comput.Phys.Commun.176:300-304,2007 doi: 10.1016/j.cpc.2006.11.010
- [8] F Beaudette for the CMS Collaboration *The CMS Particle Flow Algorithm* Proceedings of the CHEP2013 Conference, p295 (2013), ISBN 978-2-7302-1624-1
- [9] M Hildreth *A New Pileup Mixing Framework for CMS* Proceedings of this conference CHEP2015